REMARKS

This paper is responsive to a Final Office action dated April 8, 2008. Claims 1-3, 5-16, 18-22, 26-35, 37-43, 45-47, 49, 51-53, and 57 were examined.

Claim Rejections Under 35 U.S.C. § 102

Claims 33-35, 37-39, 48, 49, 51, and 53 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U. S. Patent Application Publication No. 2002/0094081 to Medvinsky (hereinafter, "Medvinsky"). Claims 33-35, 37-39, 48, 49, 51, 53 are canceled to reduce issues for appeal.

Claim Rejections Under 35 U.S.C. § 103 Over Medvinsky and Jung

Claims 1-3, 5-8, 14-16, 18-22, 26, 27, 41-43, and 45-47 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Medvinsky in view of U. S. Patent Application Publication No. 2001/0052072 to Jung (hereinafter, "Jung").

Regarding claim 1, Applicants respectfully maintain that Medvinsky, alone or in combination with Jung, fails to teach or suggest

> padding an encrypted payload of the received data packet to a given size with padding, the given size corresponding to the fixed length segment size, and decrypting the payload of the received data packet by applying the fixed length segment of the continuous decryption key to the padded, encrypted payload, a portion of the fixed length segment being applied to the encrypted payload, a remaining portion of the fixed length segment being applied to the padding,

as required by claim 1. Medvinsky teaches that

> [a]fter the encrypted data stream is received, processor 134 (of remote MTA 114) directs key stream generator 132 to output the same key stream bytes from the same key stream that was used to encrypt the voice packets at the local end. The

> key stream generator either generates the key stream bytes on-demand, or is free running based on the MTA clock and has the key stream bytes available by the time the RTP packet is received.
>
> Ncxt, packet decryptor 130 XORs the key stream bytes with the encrypted data to recover the voice packets. The RTP time stamp is always incrementing to point to a unique place in the key stream such that packet decryptor 130 recovers the encrypted data. The present invention ensures that the key stream bytes are never repeated and thus enables secure communication of voice packets, even if a CODEC change or an SSRC collision occurs as further described with reference to FIG. 2. As used hcrein a "time stamp" is any mechanism for performing synchronization for a cipher in order to attain decryption of encrypted data.

Paragraphs 0033 and 0034. Nowhere does Mcdvinsky teach or suggest <u>padding an encrypted payload of the received data packet</u> to a given size with padding, the given size corresponding to the fixed length segment size, and decrypting the payload of the received data packet by applying the fixed length segment of the continuous decryption key to the <u>padded, encrypted payload</u>, a portion of the fixed length segment being applied to the encrypted payload, a remaining portion of the fixed length segment being applied to the padding, as required by claim 1.

Jung fails to compensate for the shortcomings of Medvinsky. Jung teaches that

> [t]he encryption/decryption module 24 is primarily responsible for encrypting the outgoing speech data packets and decrypting the incoming speech data packets. In one embodiment, a stream encryption algorithm is used by the encryption/decryption module 24 to encrypt and decrypt the data packets. Note, however, that the specific type of stream encryption algorithm used is not important to the invention, and that any known or yet to be developed stream encryption may be used without departing from the scope of the invention. The tasks performed by the encryption/decryption module 24 include such things as performing certain mathematical/logical operations on the data (depending on the type of encryption used), <u>padding the data where applicable</u>, and other tasks related to the encryption/decryption process.

Paragraph 0035 (emphasis added). Padding data <u>where applicable</u>, as taught by Jung fails to teach or suggest <u>padding an encrypted payload of the received data packet</u> to a given size with padding, the given size corresponding to the fixed length segment size, and <u>decrypting</u> the payload of the received data packet by applying the fixed length segment of the continuous decryption key to the <u>padded, encrypted payload</u>, a portion of the fixed length segment being applied to the encrypted payload, a remaining portion of the fixed length segment being applied

to the padding, as required by claim 1. No other portion of Jung teaches those limitations of claim 1.

Since Medvinsky and Jung fail to teach or suggest the recited limitations and no other art of record adds the missing disclosure, Applicants respectfully request that the rejection of claim 1 and all claims dependent thereon, be withdrawn.

Regarding claim 14, Applicants respectfully maintain that Medvinsky, alone or in combination with Jung, fails to teach or suggest

> padding data to generate padded data, applying the fixed length segment to the padded data to form padded encrypted data by applying a portion of the fixed length segment to the data to form an encrypted payload and applying a remaining portion of the fixed length segment to the padding, de-padding the padded encrypted data to form the encrypted payload, and combining the encrypted payload and the at least a portion of the session count to form an encrypted data packet,

as required by claim 14. Medvinsky teaches that

> [a]fter the encrypted data stream is received, processor 134 (of remote MTA 114) directs key stream generator 132 to output the same key stream bytes from the same key stream that was used to encrypt the voice packets at the local end. The key stream generator either generates the key stream bytes on-demand, or is free running based on the MTA clock and has the key stream bytes available by the time the RTP packet is received.
>
> Next, packet decryptor 130 XORs the key stream bytes with the encrypted data to recover the voice packets. The RTP time stamp is always incrementing to point to a unique place in the key stream such that packet decryptor 130 recovers the encrypted data. The present invention ensures that the key stream bytes are never repeated and thus enables secure communication of voice packets, even if a CODEC change or an SSRC collision occurs as further described with reference to FIG. 2. As used herein a "time stamp" is any mechanism for performing synchronization for a cipher in order to attain decryption of encrypted data.

Paragraphs 0033 and 0034. Nowhere does Medvinsky teach or suggest padding data to generate padded data, applying the fixed length segment to the padded data to form padded encrypted data by applying a portion of the fixed length segment to the data to form an encrypted payload and applying a remaining portion of the fixed length segment to the padding, de-padding the padded encrypted data to form the encrypted payload, and combining the encrypted payload and the at least a portion of the session count to form an encrypted data packet, as required by claim 14.

Jung fails to compensate for the shortcomings of Medvinsky. Jung teaches that

> [t]he encryption/decryption module 24 is primarily responsible for encrypting the outgoing speech data packets and decrypting the incoming speech data packets. In one embodiment, a stream encryption algorithm is used by the encryption/decryption module 24 to encrypt and decrypt the data packets. Note, however, that the specific type of stream encryption algorithm used is not important to the invention, and that any known or yet to be developed stream encryption may be used without departing from the scope of the invention. The tasks performed by the encryption/decryption module 24 include such things as performing certain mathematical/logical operations on the data (depending on the type of encryption used), padding the data where applicable, and other tasks related to the encryption/decryption process.

Paragraph 0035 (emphasis added). Padding data where applicable, as taught by Jung fails to teach or suggest padding data to generate padded data, applying the fixed length segment to the padded data to form padded encrypted data by applying a portion of the fixed length segment to the data to form an encrypted payload and applying a remaining portion of the fixed length segment to the padding, de-padding the padded encrypted data to form the encrypted payload, and combining the encrypted payload and the at least a portion of the session count to form an encrypted data packet, as required by claim 14. No other portion of Jung teaches those limitations of claim 14.

Since Medvinsky and Jung fail to teach or suggest the recited limitations and no other art of record adds the missing disclosure, Applicants respectfully request that the rejection of claim 14 and all claims dependent thereon, be withdrawn.

Regarding claim 41, Applicants respectfully maintain that Medvinsky, alone or in combination with Jung, fails to teach or suggest

> a padding engine configured to generate padded data, an encryption engine configured to apply a portion of a fixed length segment of a continuous encryption key stream to the padded data to form encrypted padded data, a pad remover coupled to receive the encrypted padded data from the encryption engine and operable to remove the encrypted padding to generate an encrypted payload,

as required by claim 41. Medvinsky teaches that

> [a]fter the encrypted data stream is received, processor 134 (of remote MTA 114) directs key stream generator 132 to output the same key stream bytes from the same key stream that was used to encrypt the voice packets at the local end. The key stream gencrator either generates the key stream bytes on-demand, or is free running based on the MTA clock and has the key stream bytes available by the time the RTP packet is received.
>
> Next, packet decryptor 130 XORs the key stream bytes with the encrypted data to recover the voice packets. The RTP time stamp is always incrementing to point to a unique place in the key stream such that packet decryptor 130 recovers the encryptcd data. The present invention ensures that the key stream bytes are never repeated and thus enables secure communication of voice packets, even if a CODEC change or an SSRC collision occurs as further described with reference to FIG. 2. As used herein a "time stamp" is any mechanism for performing synchronization for a cipher in order to attain decryption of encrypted data.

Paragraphs 0033 and 0034. Nowhere does Medvinsky tcach or suggest a padding engine configured to generate padded data, an encryption engine configured to apply a portion of a fixed length segment of a continuous encryption key stream to the padded data to form encrypted padded data, a pad remover coupled to receive thc encrypted padded data from the encryption engine and opcrable to remove the encrypted padding to generate an encrypted payload, as required by claim 41.

Jung fails to compensate for thc shortcomings of Medvinsky. Jung teaches that

> [t]he encryption/decryption module 24 is primarily responsible for encrypting the outgoing speech data packets and decrypting the incoming speech data packets. In one embodiment, a stream encryption algorithm is used by the

> encryption/decryption module 24 to encrypt and decrypt the data packets. Note, however, that the specific type of stream encryption algorithm used is not important to the invention, and that any known or yet to be developed stream encryption may be used without departing from the scope of the invention. The tasks performed by the encryption/decryption module 24 include such things as performing certain mathematical/logical operations on the data (depending on the type of encryption used), <u>padding the data where applicable</u>, and other tasks related to the encryption/decryption process.

Paragraph 0035 (emphasis added). Padding data <u>where applicable</u>, as taught by Jung fails to teach or suggest <u>a padding engine</u> configured to generate <u>padded data</u>, an encryption engine configured to apply a portion of a fixed length segment of a continuous encryption key stream to the padded data to form <u>encrypted padded data</u>, <u>a pad remover</u> coupled to receive the encrypted padded data from the encryption engine and operable to remove the encrypted padding to generate an <u>encrypted payload</u>, as required by claim 41. No other portion of Jung teaches those limitations of claim 41.

Since Medvinsky and Jung fail to teach or suggest the recited limitations and no other art of record adds the missing disclosure, Applicants respectfully request that the rejection of claim 41 and all claims dependent thereon, be withdrawn.

<u>Claim Rejections Under 35 U.S.C. § 103 Over Medvinsky, Jung, and Chang</u>

Claims 9-13 and 28-32 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Medvinsky in view of Jung, and further in view of U.S. Patent No. 6,104,012 to Chang et al. (hereinafter, "Chang").

Regarding claims 9-13, Applicants respectfully maintain that Medvinsky, alone or in combination with Jung and Chang, fails to teach or suggest

> <u>padding an encrypted payload of the received data packet</u> to a given size with padding, the given size corresponding to the fixed length segment size, and decrypting the payload of the received data packet by applying the fixed length segment of the continuous decryption key to the <u>padded, encrypted payload</u>, a

> portion of the fixed length segment being applied to the encrypted payload, a remaining portion of the fixed length segment being applied to the padding,

as required by claim 1, from which claims 9-13 depend. As discussed above with regard to claim 1, Medvinsky and Jung fail to teach or suggest padding an encrypted payload of the received data packet to a given size with padding, the given size corresponding to the fixed length segment size, and decrypting the payload of the received data packet by applying the fixed length segment of the continuous decryption key to the padded, encrypted payload, a portion of the fixed length segment being applied to the encrypted payload, a remaining portion of the fixed length segment being applied to the padding, as required by claim 1.

Sengodan fails to compensate for the shortcomings of Medvinsky and Jung. Sengodan teaches that "[t]he recipient after decrypting the mini-packet looks at the last byte 524 to determine the number of padding bytes 522 used." Col. 8, lines 19-21. Nowhere does Sengodan teach or suggest padding any portion of a received data packet, as required by claim 1.

Since Medvinsky, Jung, and Sengodan fail to teach or suggest the recited limitations and no other art of record adds the missing disclosure, Applicants respectfully request that the rejection of claims 9-13 be withdrawn.

Regarding claims 28-32, Applicants respectfully maintain that Medvinsky, alone or in combination with Jung and Chang, fails to teach or suggest

> padding data to generate padded data, applying the fixed length segment to the padded data to form padded encrypted data by applying a portion of the fixed length segment to the data to form an encrypted payload and applying a remaining portion of the fixed length segment to the padding, de-padding the padded encrypted data to form the encrypted payload, and combining the encrypted payload and the at least a

> portion of the session count <u>to form an encrypted data packet</u>,

as required by claim 14 from which claims 28-32 depend. As discussed above with regard to claim 14, Medvinsky and Jung fail to teach or suggest padding data to <u>generate padded data</u>, applying the fixed length segment to the padded data to form <u>padded encrypted data</u> by applying a portion of the fixed length segment to the data to form an encrypted payload and applying a remaining portion of the fixed length segment to the padding, <u>de-padding the padded encrypted data to form the encrypted payload</u>, and combining the encrypted payload and the at least a portion of the session count <u>to form an encrypted data packet</u>, as required by claim 14.

Sengodan fails to compensate for the shortcomings of Medvinsky and Jung. Sengodan teaches

> assembling mini-packets into a payload wherein each mini-packet includes an associated mini-header for ensuring proper processing of each mini-packet and adding padding to mini-packets when the mini-packets are encrypted to insure each mini-packet is an integral multiple of a predetermined block size.

Col. 4, lines 30-36. Adding padding to mini-packets when the mini-packets of Sengodan are encrypted fails to teach or suggest <u>generating padded data</u>, forming <u>padded encrypted data</u>, <u>de-padding the padded encrypted data to form the encrypted payload</u>, and combining the encrypted payload and the at least a portion of the session count <u>to form an encrypted data packet</u> as required by claim 14. Nowhere does Sengodan teach or suggest those limitations of claim 14.

Since Medvinsky, Jung, and Sengodan fail to teach or suggest the recited limitations and no other art of record adds the missing disclosure, Applicants respectfully request that the rejection of claims 28-32 be withdrawn.

<u>*Claim Rejections Under 35 U.S.C. § 103 Over Medvinsky and Chang*</u>

Claims 40 and 52 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Medvinsky in view of Chang. Claims 40 and 52 are canceled to reduce issues for appeal.

Claim Rejections Under 35 U.S.C. § 103 Over Medvinsky and Sengodan

Claim 57 is amended to be in independent form. Claim 57 stands rejected under 35 U.S.C. 103(a) as being unpatentable over Medvinsky and further in view of U.S. Patent No. 6,918,034 to Sengodan et al. (hereinafter, "Sengodan").

Regarding claim 57, in the Response to Arguments of the final Office action, the Office states that the limitations of claim 57 are well known, however, the Office fails to point out where the references of record teach or suggest the limitations of claim 57. Applicants respectfully maintain that Medvinsky, alone or in combination with Sengodan, fails to teach or suggest

> a receiver including a padding engine operable to pad an encrypted payload of the received encrypted data packet to generate the padded encrypted payload of the received encrypted data received by the decryption engine, a decryption engine configured to decrypt the padded encrypted payload of the received encrypted data packet by applying a portion of a current fixed length segment of a continuous decryption key stream to the padded encrypted payload to generate the padded decrypted data if the difference is less than the threshold, and a pad remover configured to remove padding from the padded decrypted data to recover the data,

as required by claim 57. Medvinsky teaches that

> [a]fter the encrypted data stream is received, processor 134 (of remote MTA 114) directs key stream generator 132 to output the same key stream bytes from the same key stream that was used to encrypt the voice packets at the local end. The key stream generator either generates the key stream bytes on-demand, or is free running based on the MTA clock and has the key stream bytes available by the time the RTP packet is received.

> Next, packet decryptor 130 XORs the key stream bytes with the encrypted data to recover the voice packets. The RTP time stamp is always incrementing to point to a unique place in the key stream such that packet decryptor 130 recovers the encrypted data. The present invention ensures that the key stream bytes are never repeated and thus enables secure communication of voice packets, even if a CODEC change or an SSRC collision occurs as further described with reference to FIG. 2. As used herein a "time stamp" is any mechanism for performing synchronization for a cipher in order to attain decryption of encrypted data.

Paragraphs 0033 and 0034. Nowhere does Medvinsky teach or suggest <u>a receiver</u> including <u>a padding engine operable to pad an encrypted payload of the received encrypted data packet</u> to generate the padded encrypted payload of the received encrypted data received by the decryption engine, <u>a decryption engine</u> configured to decrypt the padded encrypted payload of the received encrypted data packet by applying a portion of a current fixed length segment of a continuous decryption key stream to the padded encrypted payload to generate the padded decrypted data if the difference is less than the threshold, and <u>a pad remover</u> configured to remove padding from the padded decrypted data to recover the data, as required by claim 57.

Sengodan fails to compensate for the shortcomings of Medvinsky. Sengodan teaches that "[t]he recipient after decrypting the mini-packet looks at the last byte 524 to determine the number of padding bytes 522 used." Col. 8, lines 19-21. Nowhere does Sengodan teach or suggest that a <u>receiver</u> includes <u>a padding engine operable to pad an encrypted payload of the received encrypted data packet</u> to generate the padded encrypted payload of the received encrypted data received by the decryption engine, <u>a decryption engine</u> configured to decrypt the padded encrypted payload of the received encrypted data packet by applying a portion of a current fixed length segment of a continuous decryption key stream to the padded encrypted payload to generate the padded decrypted data if the difference is less than the threshold, and <u>a pad remover</u> configured to remove padding from the padded decrypted data to recover the data, as required by claim 57.

Since Medvinsky and Sengodan fail to teach or suggest the recited limitations and no other art of record adds the missing disclosure, Applicants respectfully request that the rejection of claim 57 be withdrawn.

Additional Remarks

Claim 29 is amended to correct a typographical error.

In summary, all claims are believed to be allowable over the art of record, and a Notice of Allowance to that effect is respectfully solicited. Nonetheless, if any issues remain that could be more efficiently handled by telephone, the Examiner is requested to call the undersigned at the number listed below.

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that, on the date shown below, this correspondence is being

☐ deposited with the US Postal Service with sufficient postage as first class mail in an envelope addressed as shown above.

☐ facsimile transmitted to the USPTO.

☒ transmitted using the USPTO electronic filing system.

Nicole Teitler Cave 6/9/08

Nicole Teitler Cave Date

EXPRESS MAIL LABEL: ______________________

Respectfully submitted,

Nicole Teitler Cave, Reg. No. 54,021
Attorney for Applicant(s)
(512) 338-6315 (direct)
(512) 338-6300 (main)
(512) 338-6301 (fax)